

FIG. 1

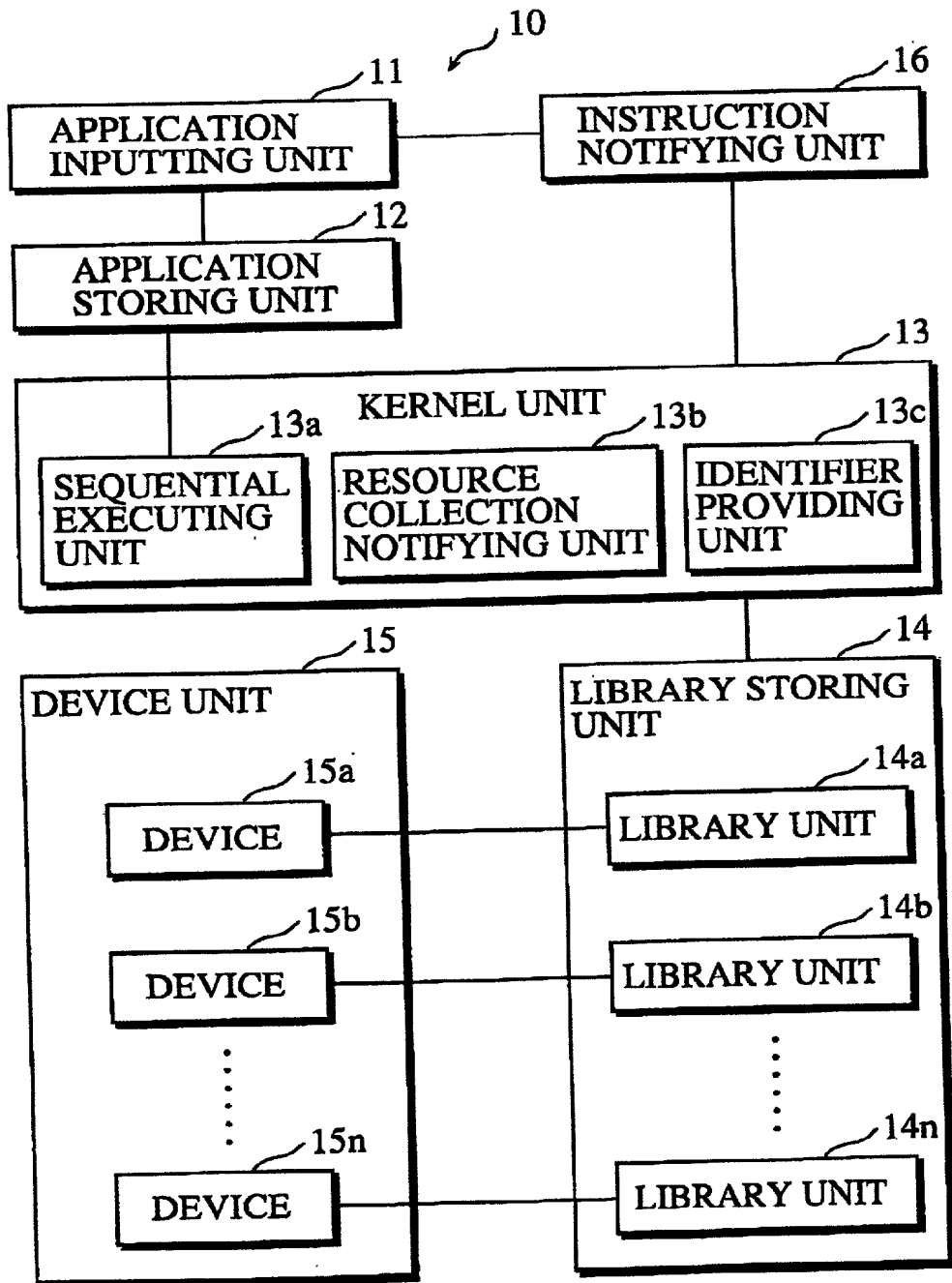


FIG. 2

```
typedef void(*CALLBACK_f)(int Appid)
```

FIG. 3

```
1  CALLBACK_f callf[100];
2
3  void init() {
4      for (I=0; I<100; I++ ) callf[I]=null;
5  }
6  int add_callf(CALLBACK_f callfunc){
7      for (I=0; I<100; I++){
8          if ( callf[I]==null) {
9              callf[I] = callfunc;
10             return(1);
11         }
12     }
13     return(-1);
14 }
15
16 void remove_callf(CALLBACK_f callfunc){
17     for (I=0; I<100; I++){
18         if ( callf[I]==callfunc ) callf[I] = null;
19     }
20 }
21
22 void call_callback (int Appid) {
23     for(I=0; I<100; I++){
24         if ( (callf[I] !=null) (*callf[I])(Appid);
25     }
26 }
```

FIG. 4

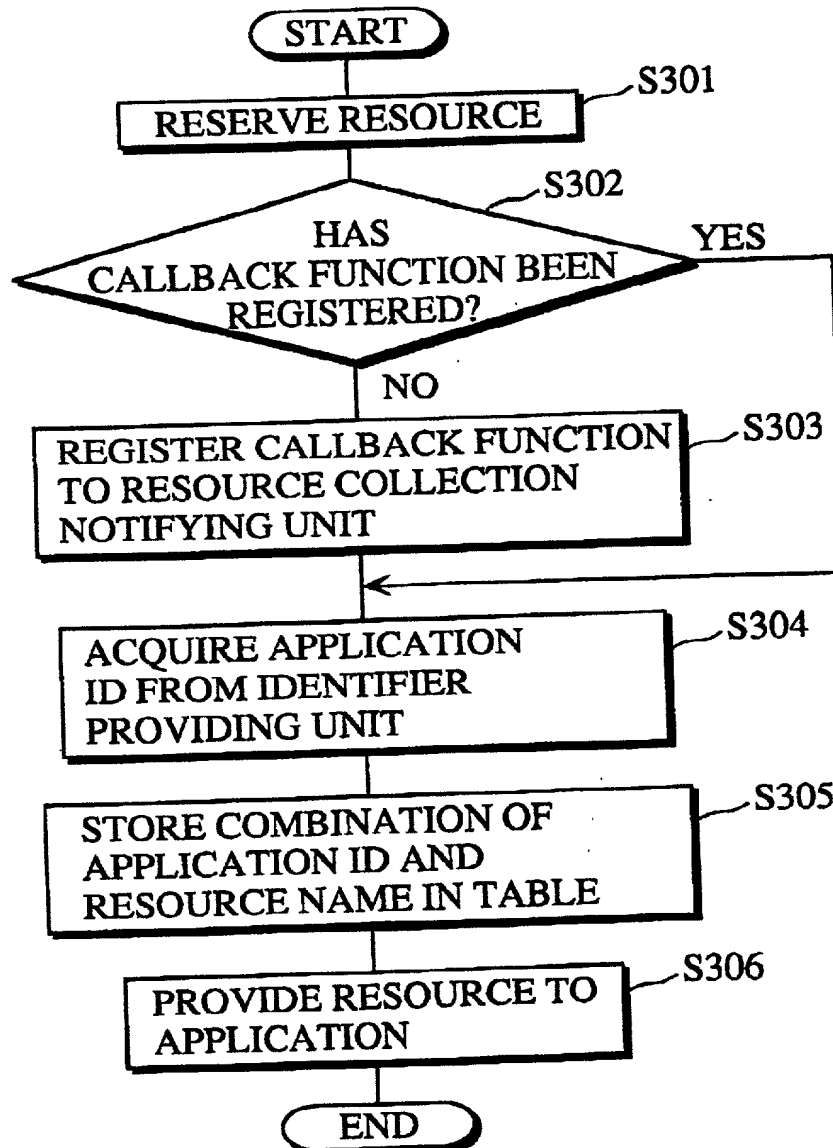


FIG. 5

(1)

APPLICATION ID	FILENAME
1	a.txt

(2)

APPLICATION ID	FILENAME
1	a.txt
2	b.txt

(3)

APPLICATION ID	FILENAME
1	a.txt

FIG. 6

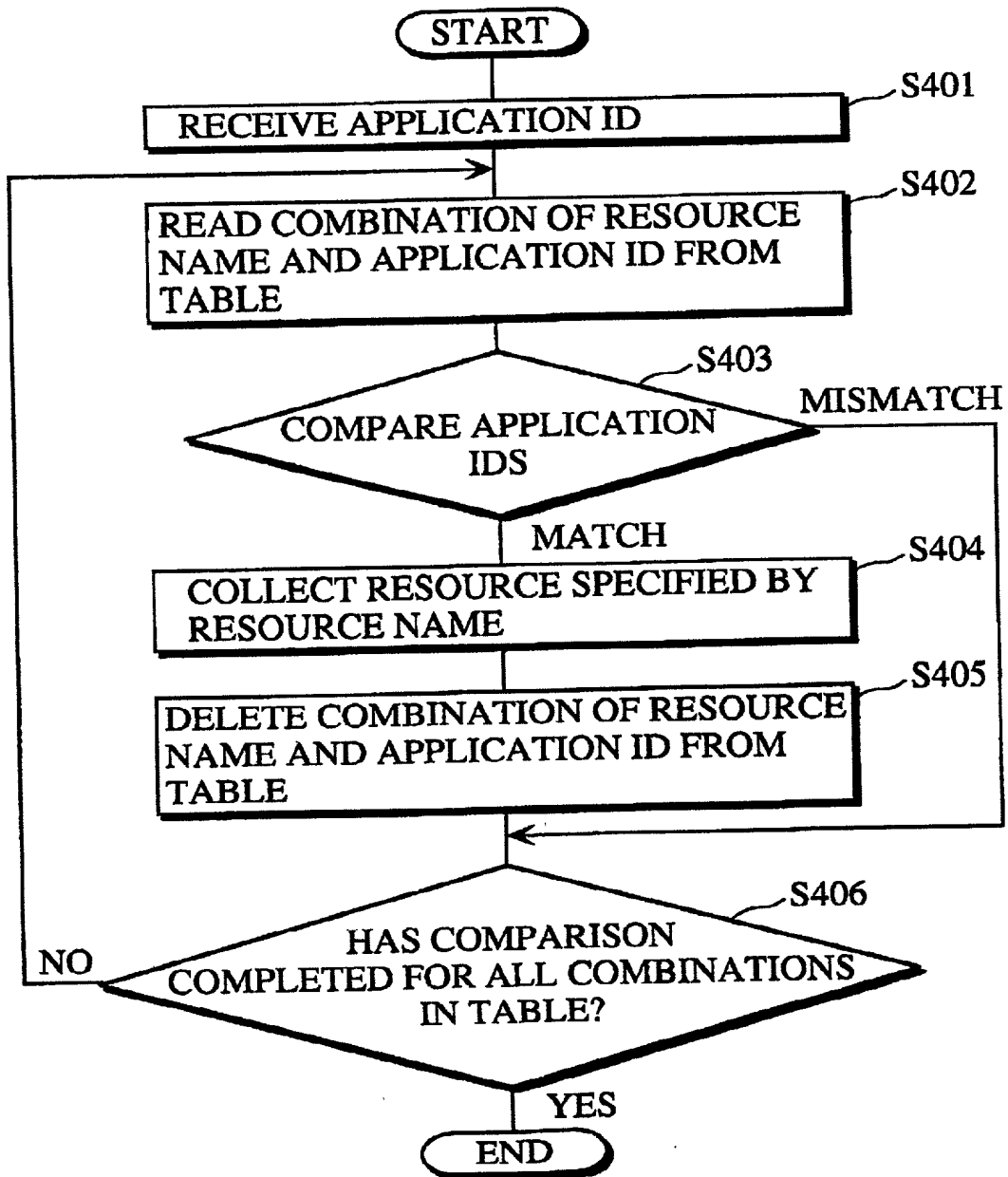


FIG. 7

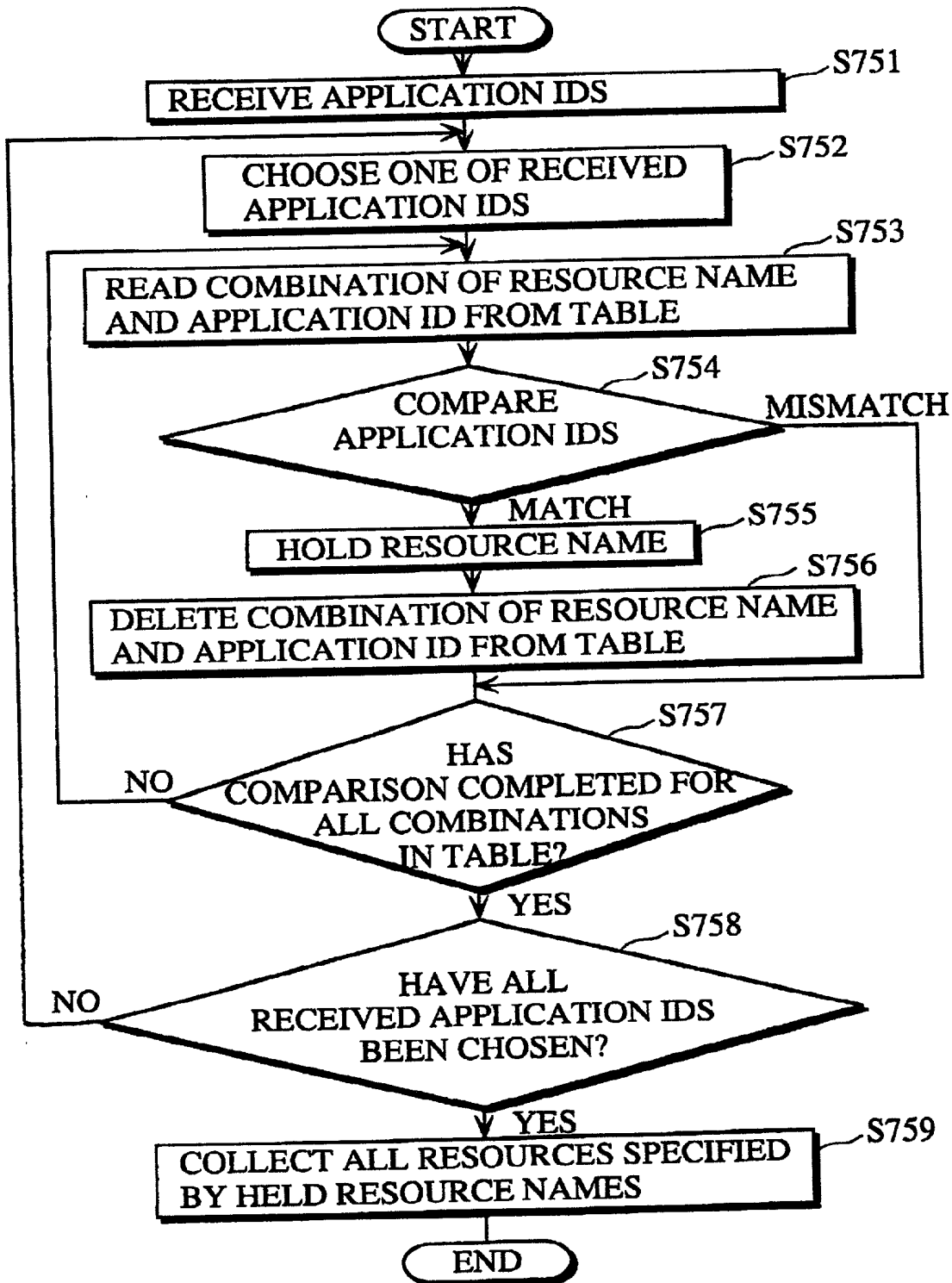


FIG. 8

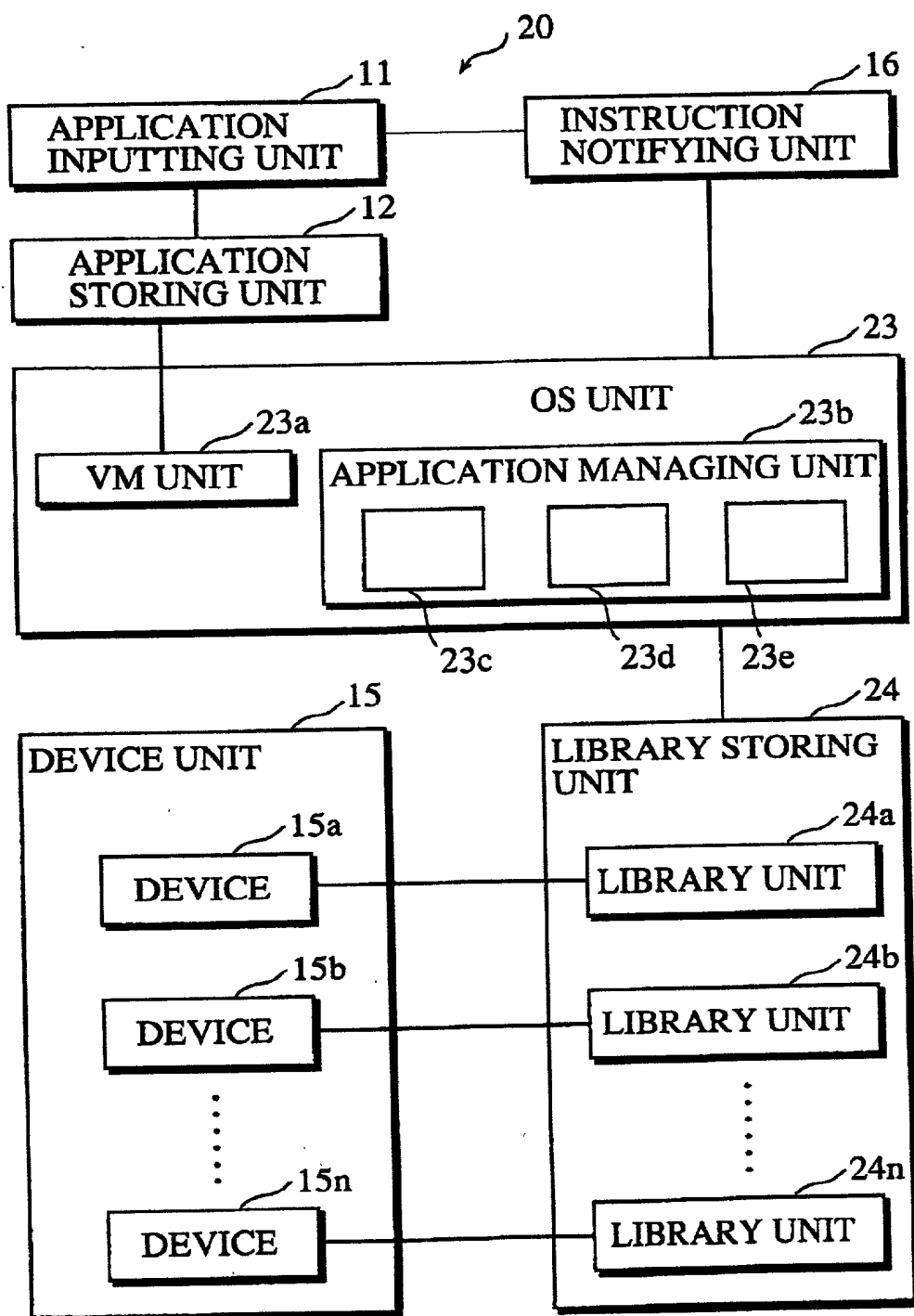


FIG. 9

```
public interface resourceCollectionListener{
    public void update();
}
```

FIG. 10

```
public class applicationProxy {
    public string name; // application name

    public int getStatus() {
        // get the status of application,paues or destroy
    }

    public addListener(resourceCollectionListener 1){
        // register resource collection instance
    }

    public removeListener(resourceCollectionListener 1){
        // remove resource collection intance
    }
}
```

FIG. 11

```
public class fileCollectionListener
    implements resourceCollectionListener{
    String name; // filename
    ApplicationProxy app;
        // Application information instance

    public void update() {
        // release file which has filename
    }
}
```


FIG. 12

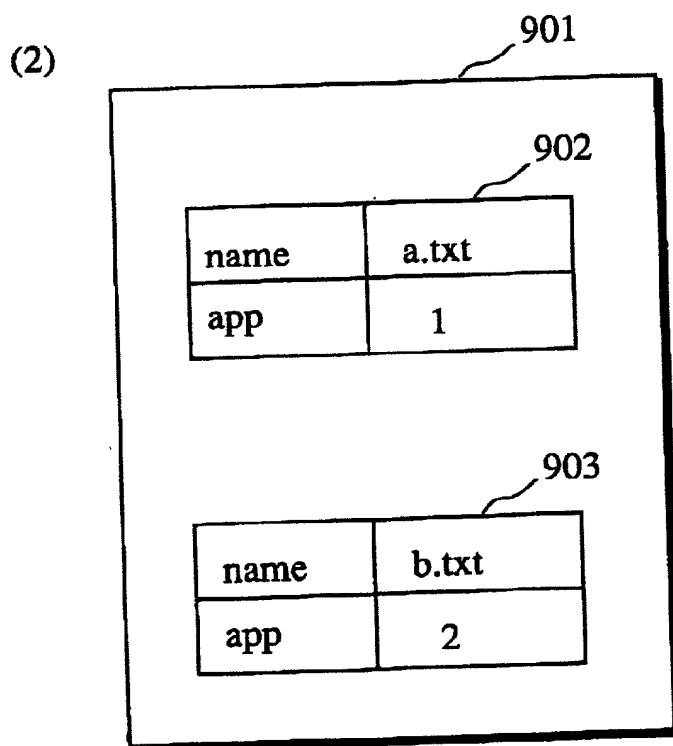
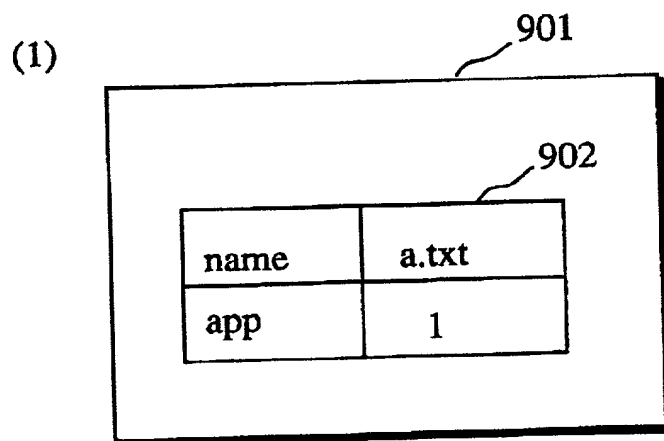


FIG. 13

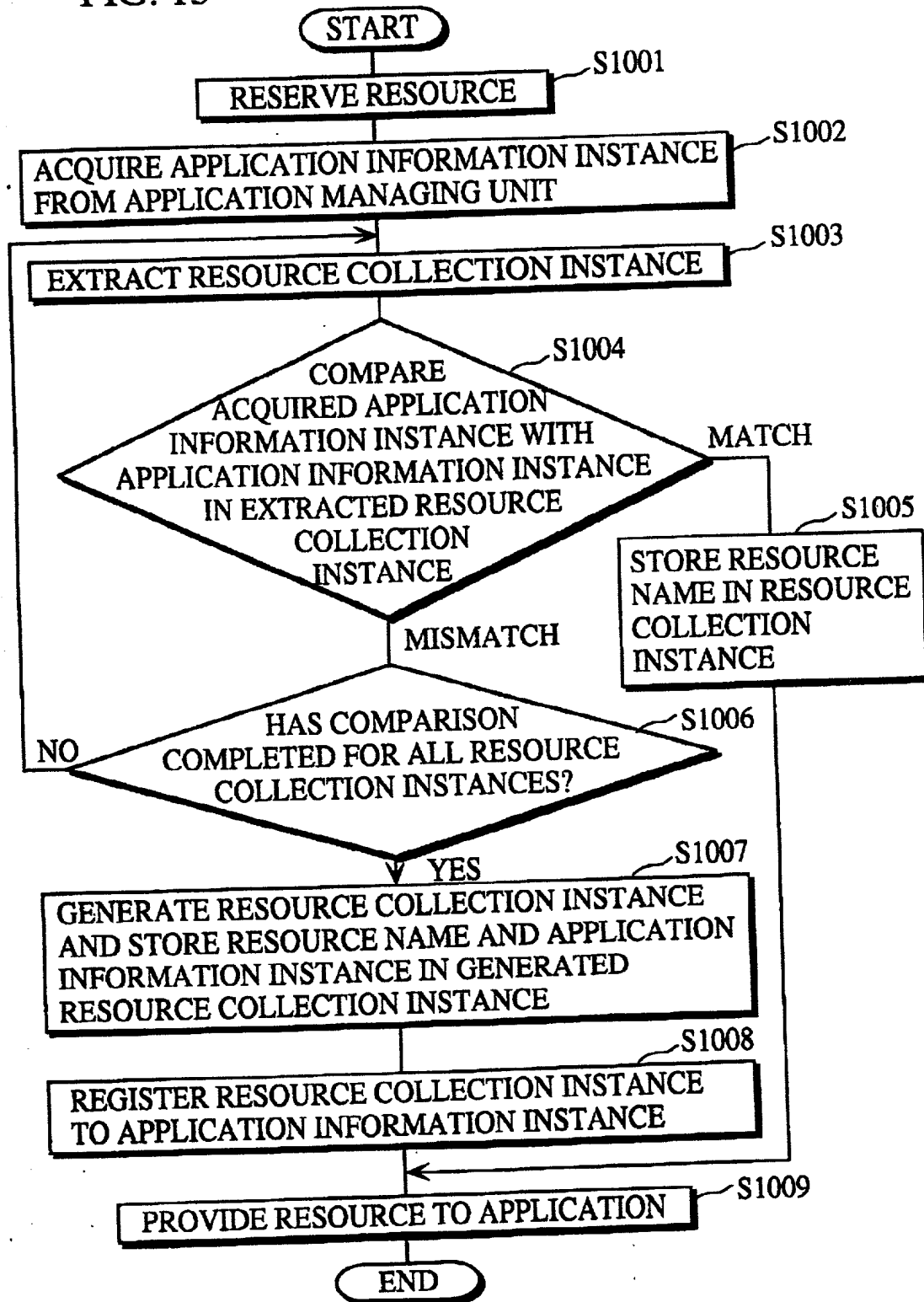


FIG. 14

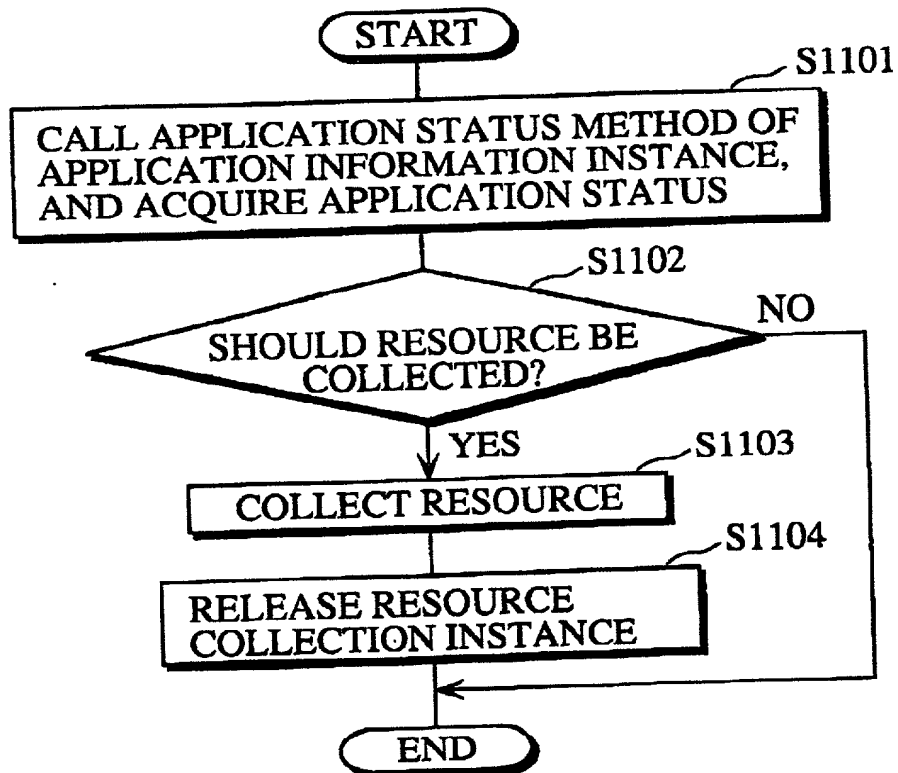


FIG. 15

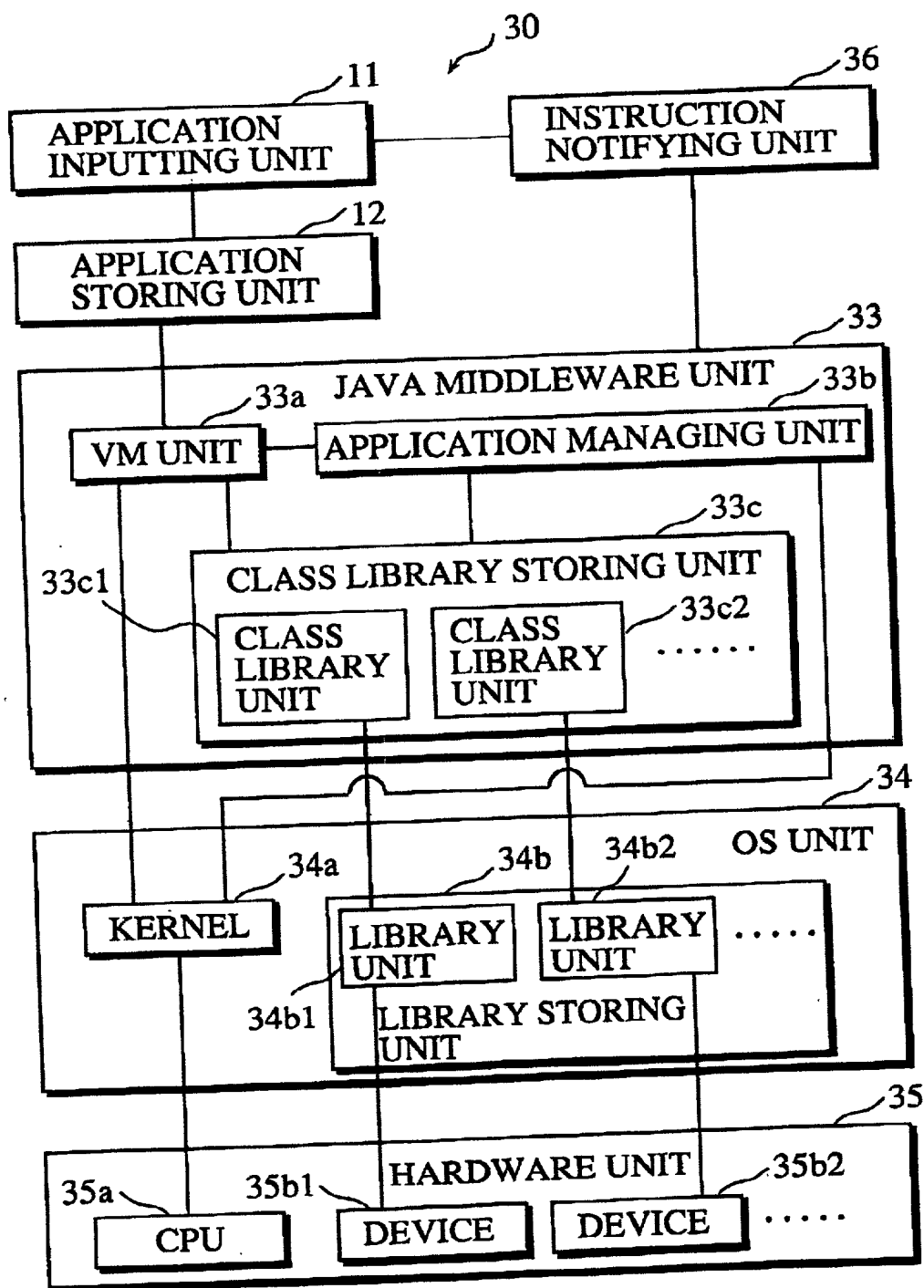


FIG. 16

(1)

APPLICATION ID	TASK ID	THREAD ID
1	201	1,2
2	202	4,5,6

(2)

APPLICATION ID	TASK ID	THREAD ID
1	201	1,2,7
2	202	4,5,6

(3)

APPLICATION ID	TASK ID	THREAD ID
1	201	1,2,7
2	202	4,6

FIG. 17

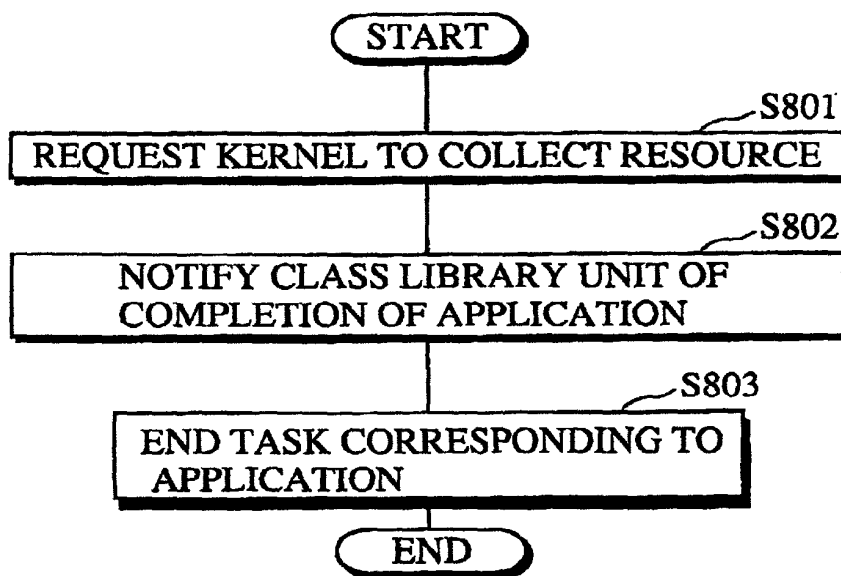


FIG. 18

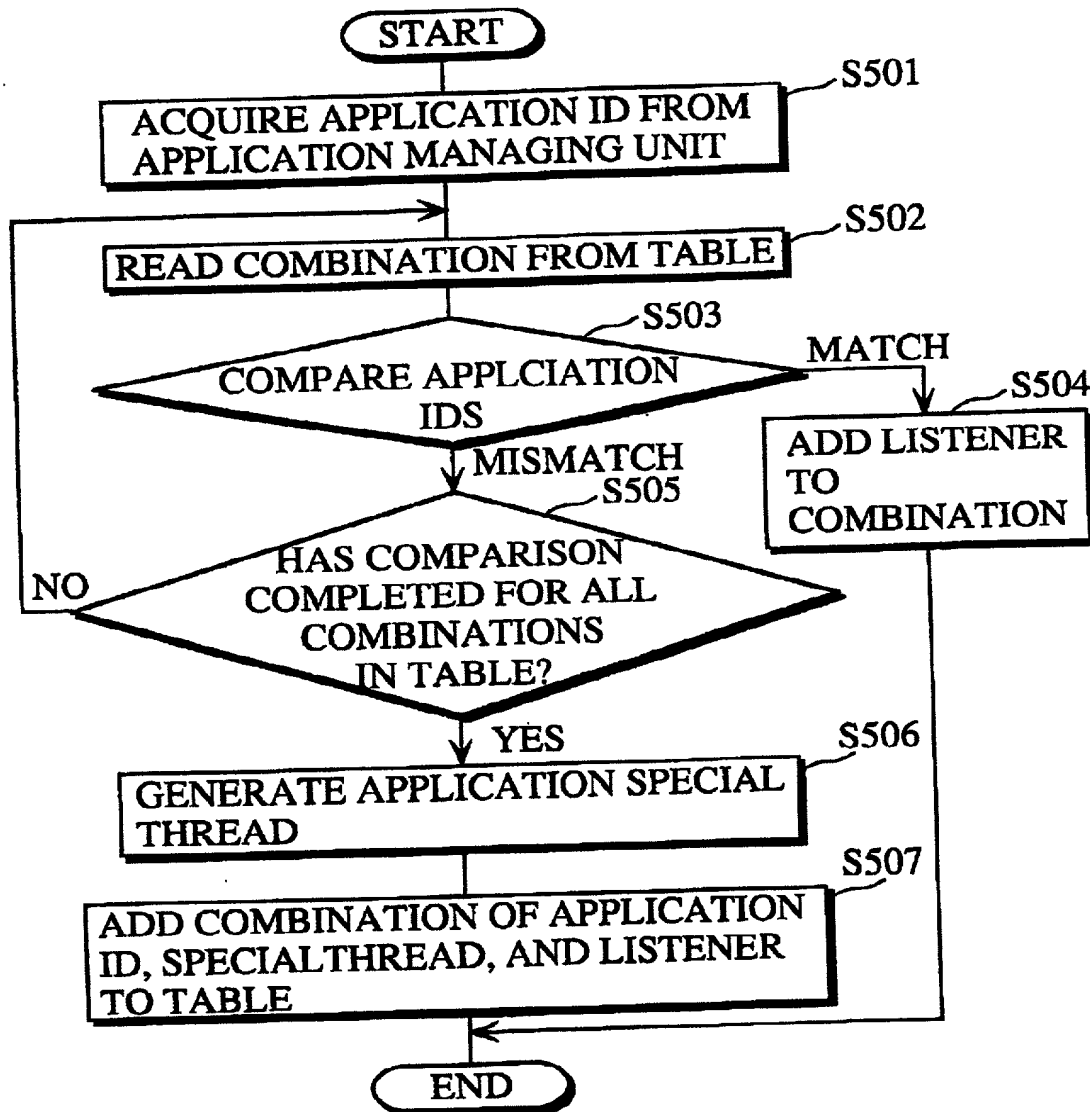


FIG. 19

(1)

APPLICATION ID	THREAD INSTANCE	LISTENER INSTANCE
1	thread10	L1,L4

(2)

APPLICATION ID	THREAD INSTANCE	LISTENER INSTANCE
1	thread10	L1,L4
2	thread20	L6

(3)

APPLICATION ID	THREAD INSTANCE	LISTENER INSTANCE
2	thread20	L6

FIG. 20

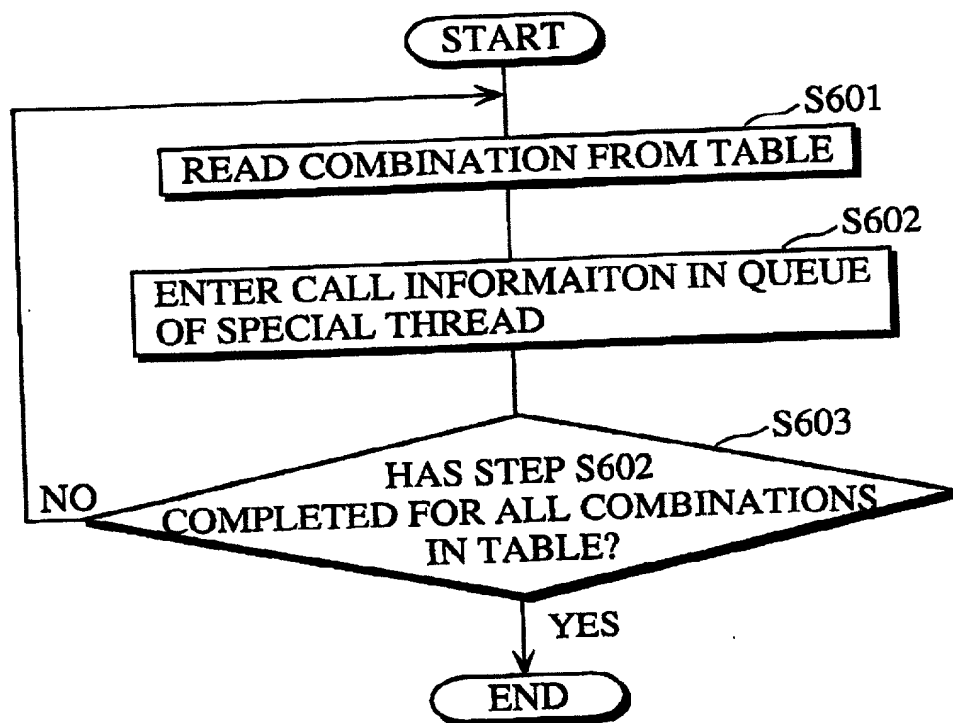


FIG. 21

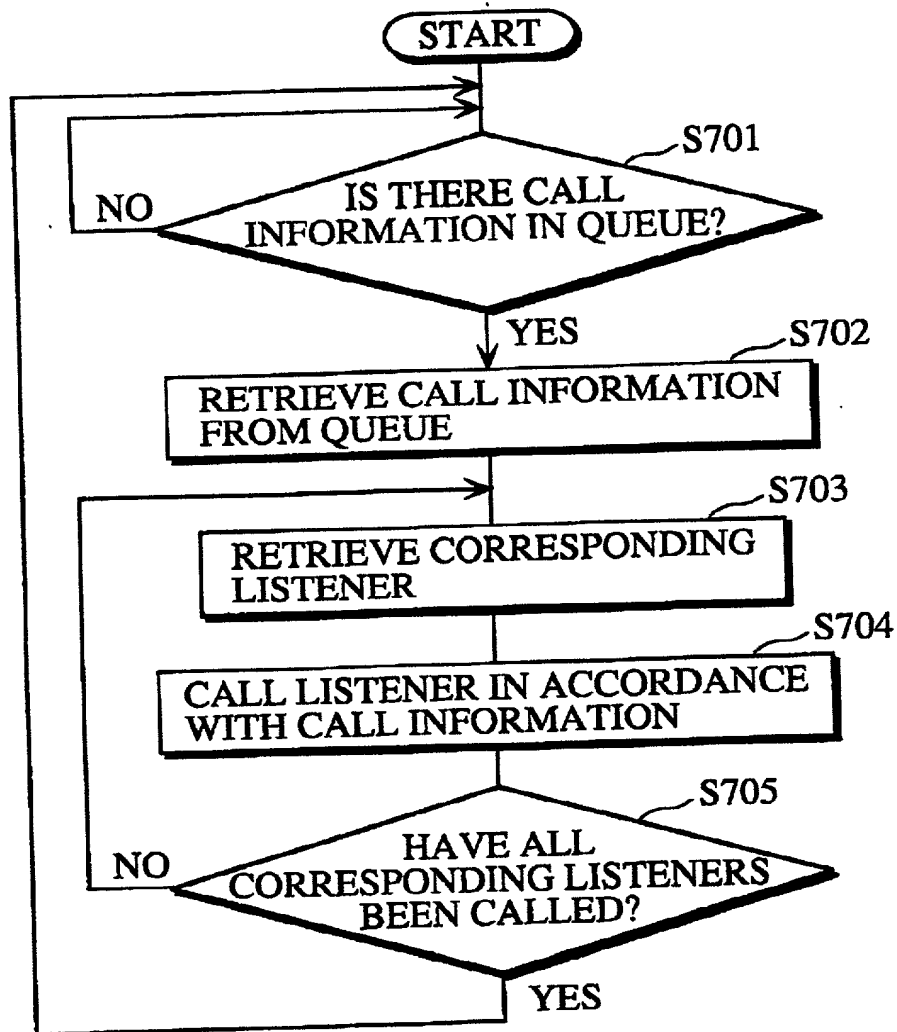


FIG. 22

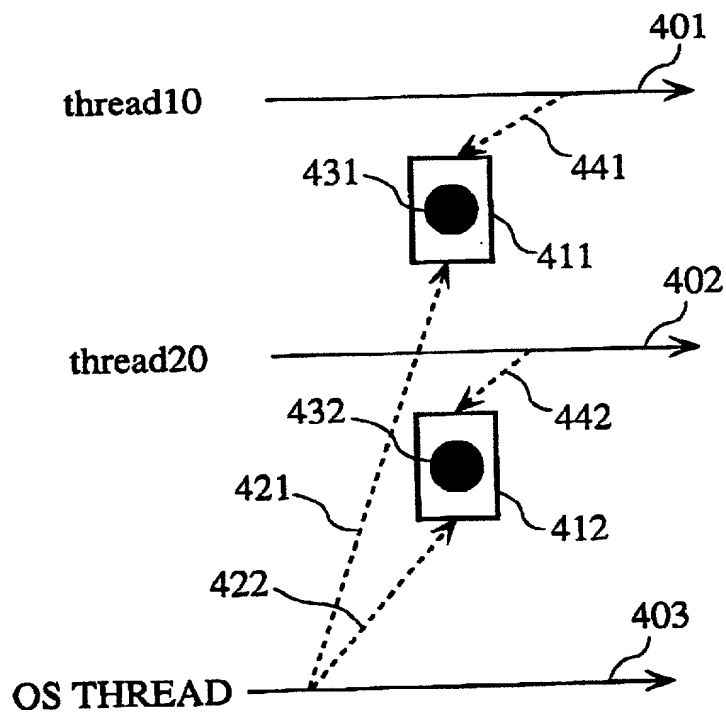


FIG. 23

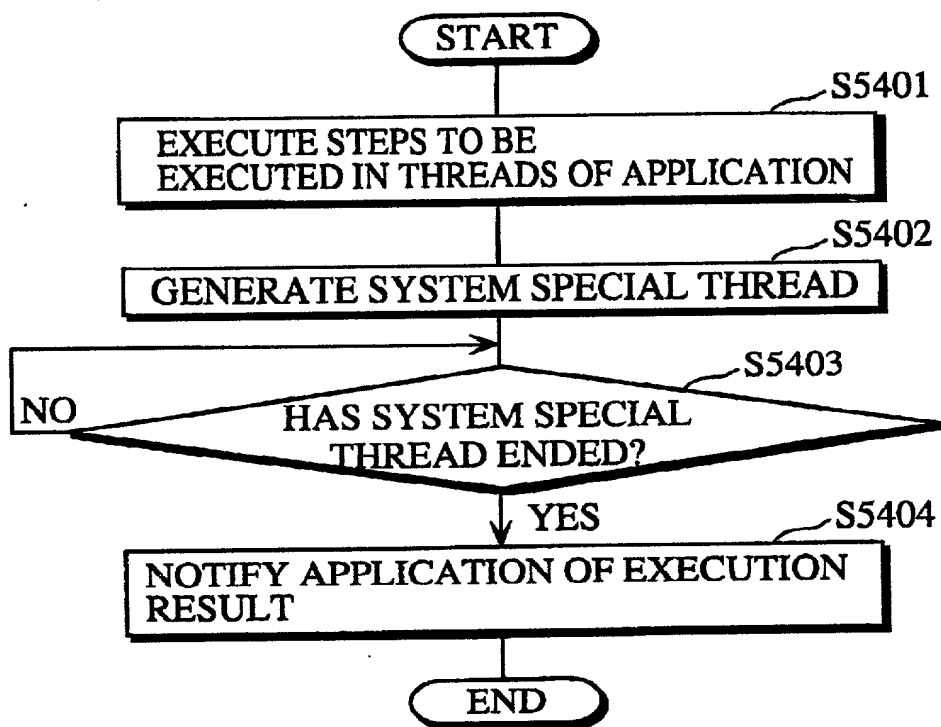


FIG. 24

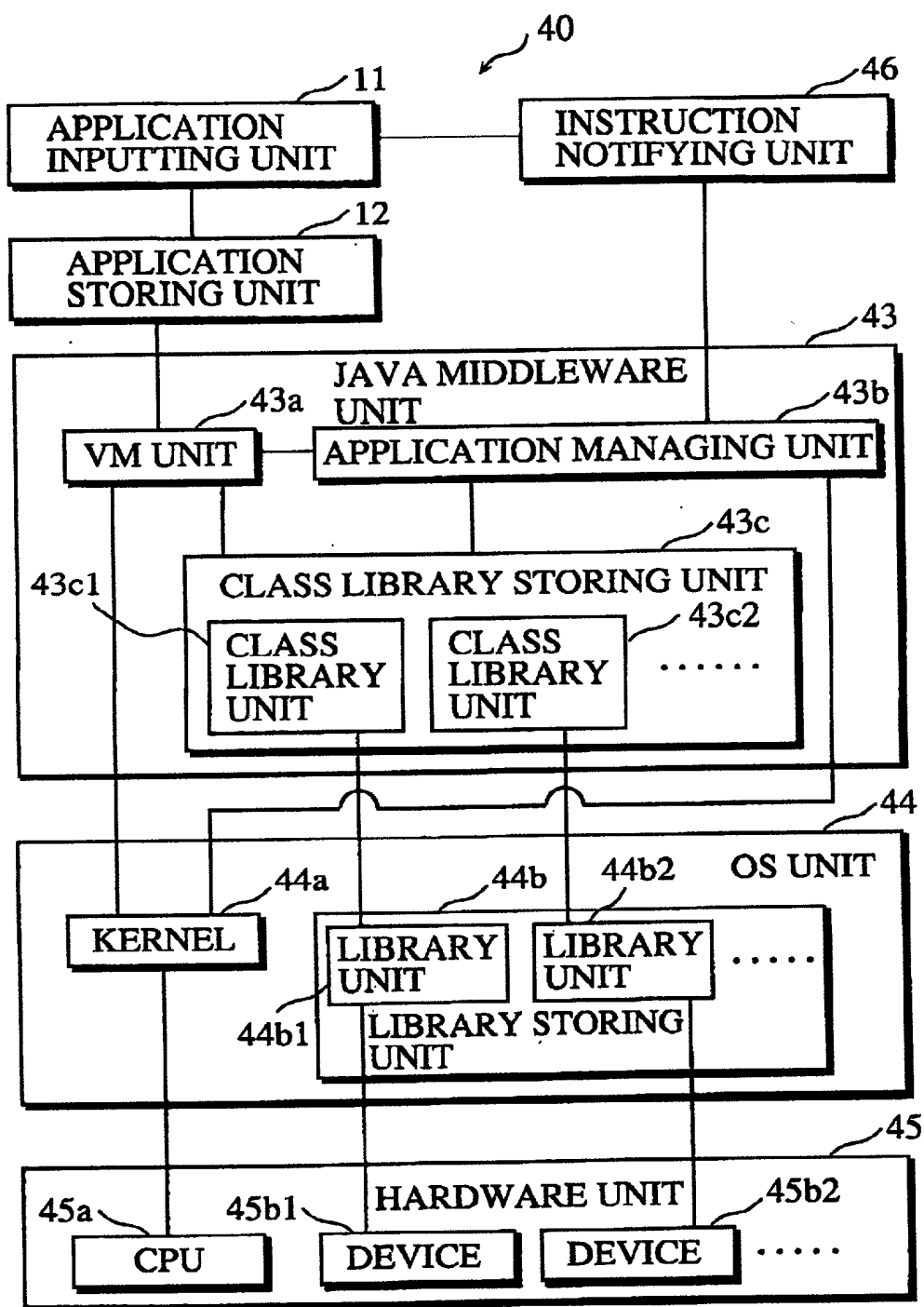


FIG. 25

APPLICATION ID	CLASS LOADER
1	classloader1
2	classloader2

FIG. 26

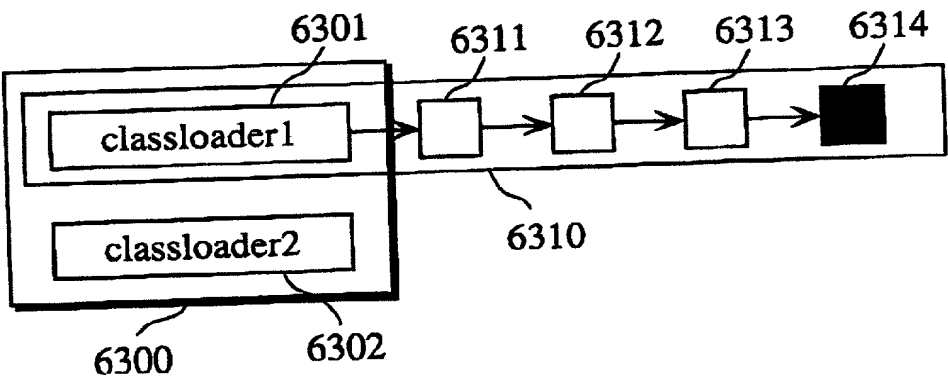


FIG. 27

APPLICATION ID	FILE
1	A.txt B.txt
2	C.txt

FIG. 28

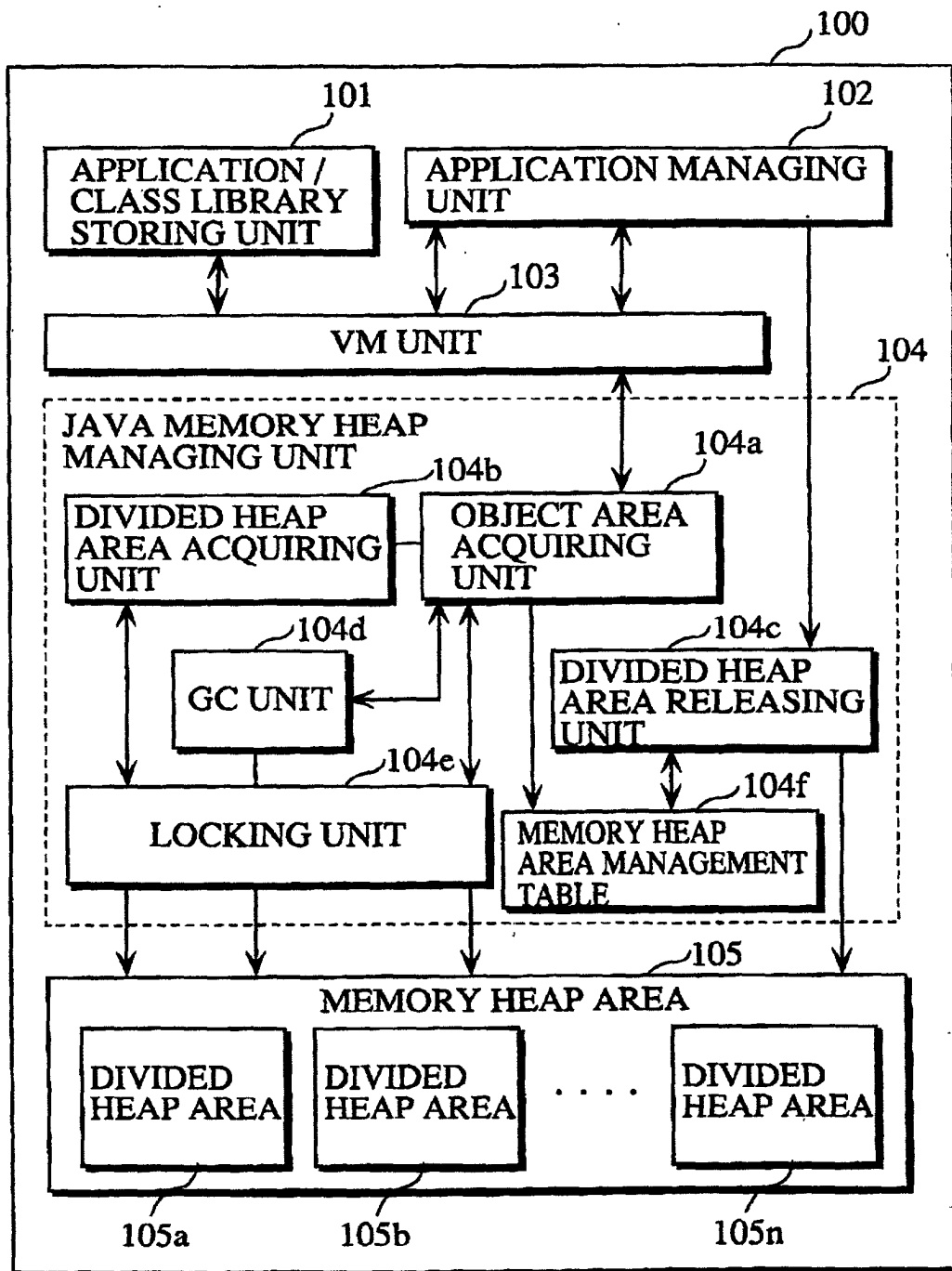


FIG. 29

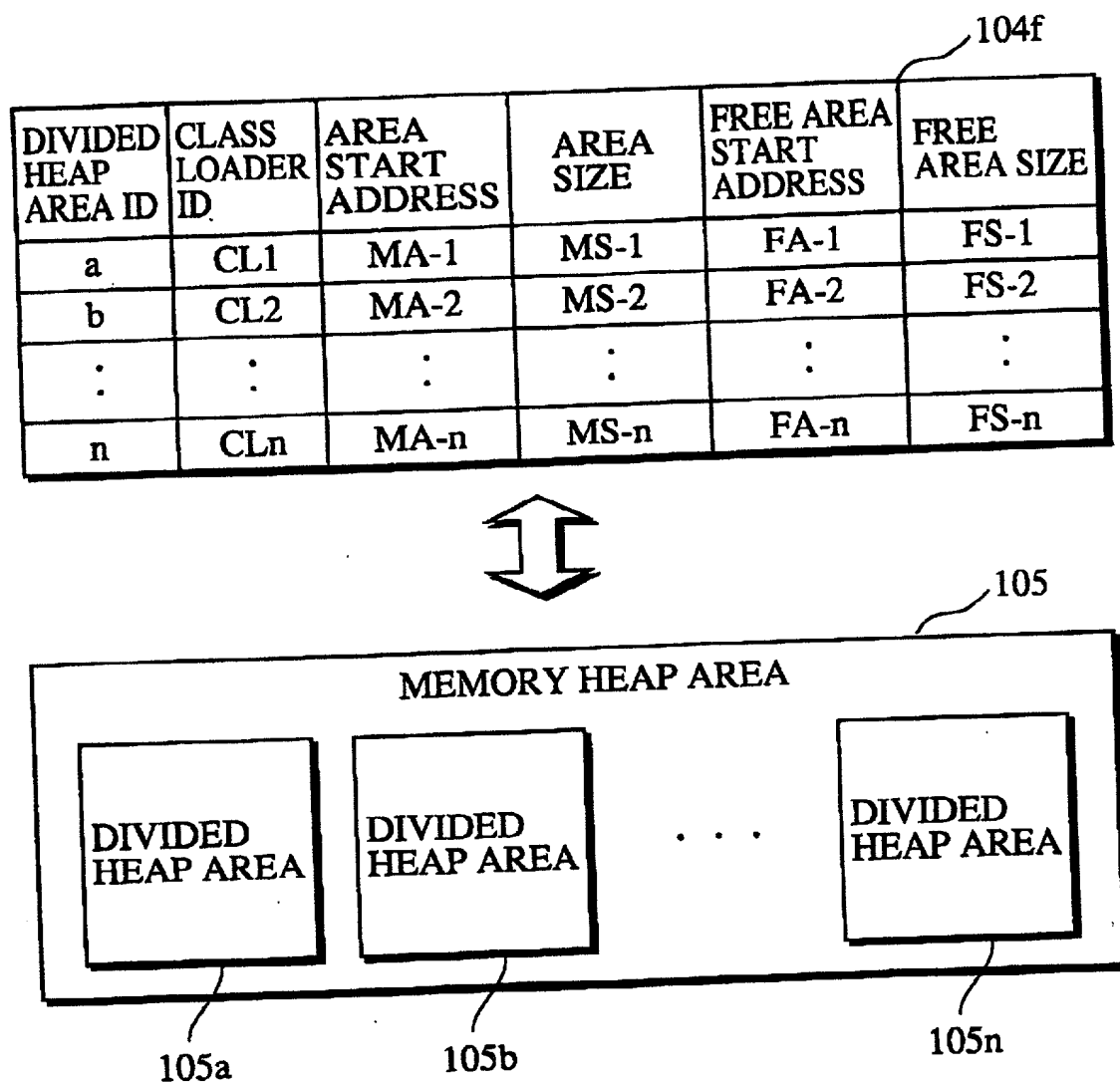


FIG. 30

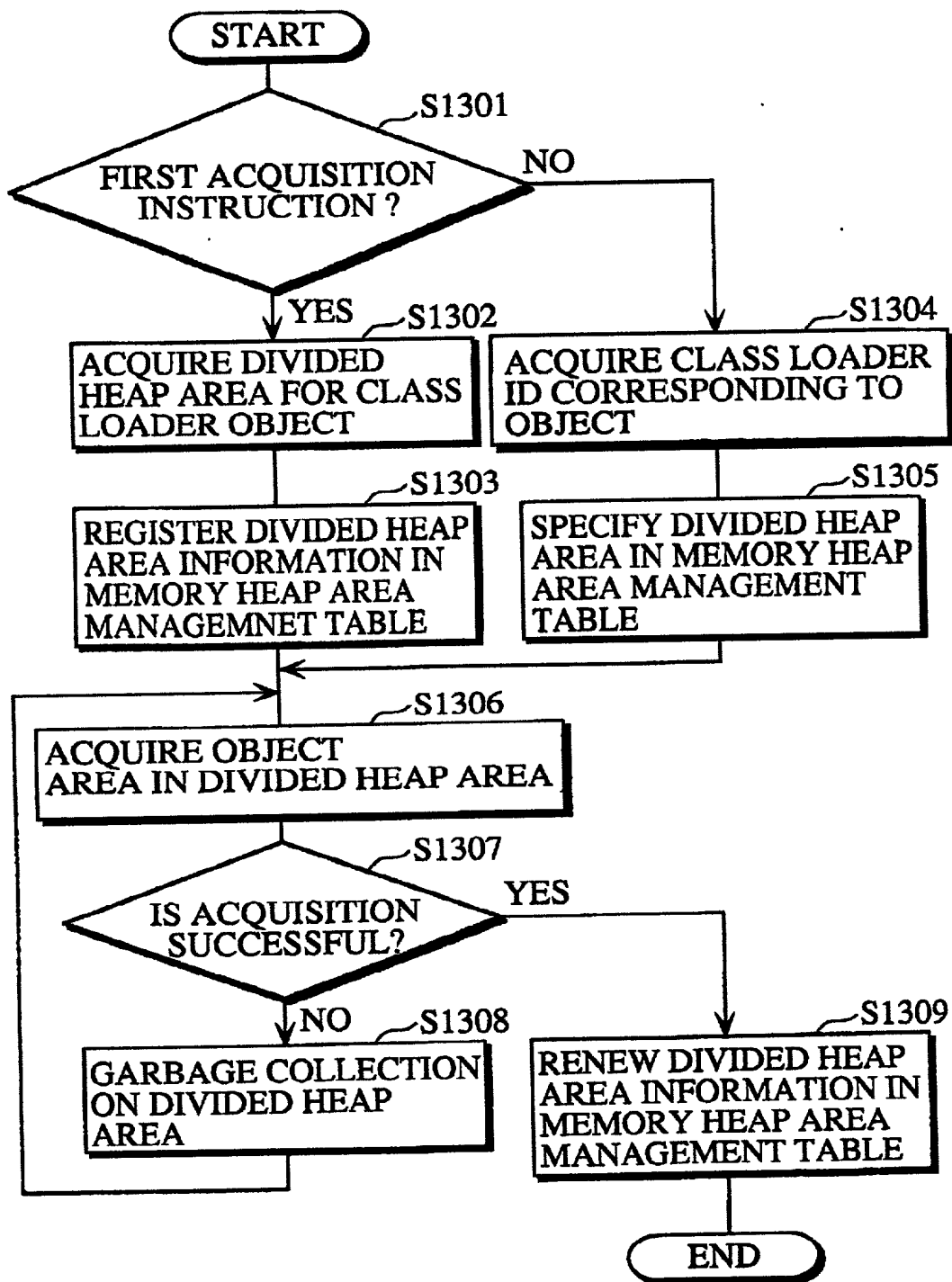


FIG. 31

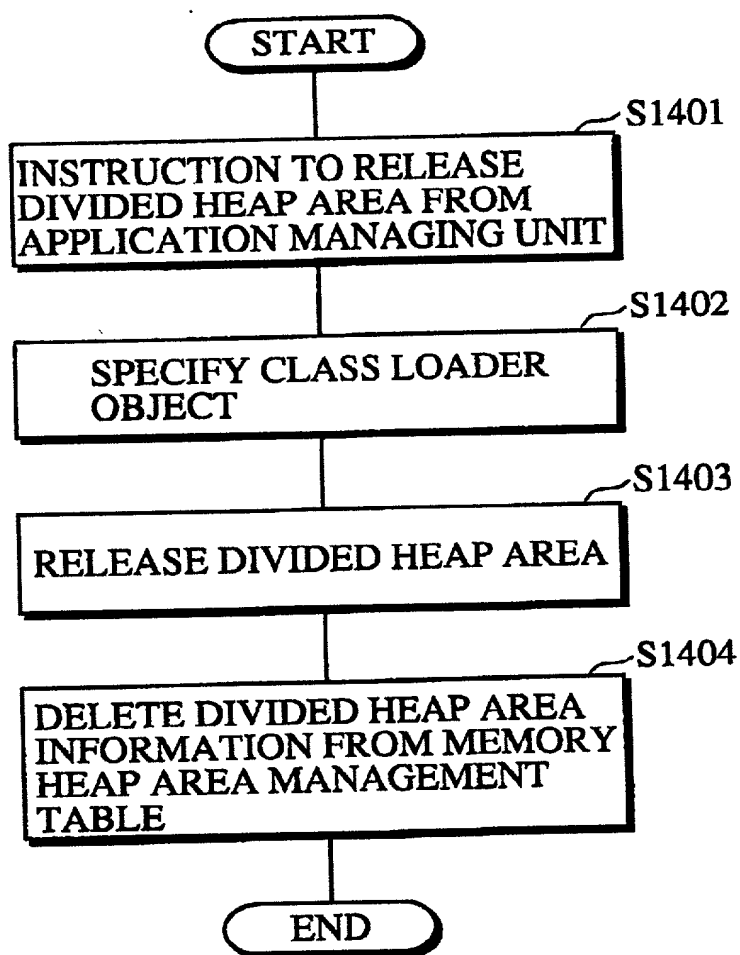


FIG. 32

DIVIDED HEAP AREA ID	CLASS LOADER ID	AREA START ADDRESS	AREA SIZE	FREE AREA START ADDRESS	FREE AREA SIZE
a0	SCL	MA-0	MS-0	FA-0	FS-0
a	CL1	MA-1	MS-1	FA-1	FS-1
:	:	:	:	:	:
n	CLn	MA-n	MS-n	FA-n	FS-n

